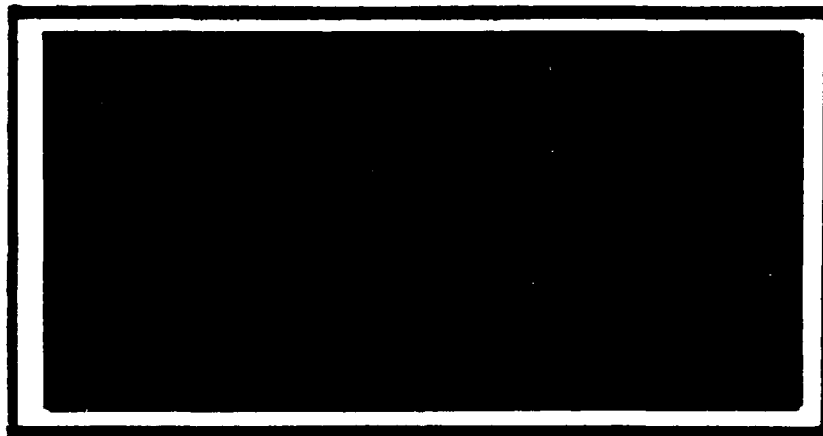




AD-A220 287



DTIC
Approved for public release;
Distribution Unlimited

DTIC
APR 09 1990
Co E

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/90M-2

COPIES OF FLOPPY DISKETTE WITH SAME TITLE
WILL BE FURNISHED UPON REQUEST BY AFIT,
ATTN: AFIT/ENS, COL. JOHN VALUSEK,
WPAFB, OH 45433-6523
TELECON 4/9/90 VG

A-1	
Dist	
Special	

A DECISION SUPPORT SYSTEM
FOR FLYING SCHEDULING IN
COMBAT COMMANDERS SCHOOL

THESIS

Javaid Anwar
Squadron Leader, PAF

AFIT/GOR/ENS/90M-2



APR 08 1990

Approved for public release; distribution unlimited

AFIT/GOR/ENS/90M-2

A DECISION SUPPORT SYSTEM FOR FLYING SCHEDULING IN
COMBAT COMMANDERS SCHOOL

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In partial Fulfillment of the Requirements for the degree of
Master of Science In Operations Research

Javaid Anwar, B.Sc.

Squadron Leader, PAF

March 90

Approved for public release; distribution unlimited

Preface

The purpose of this research was to develop a computer based scheduling system that is expected to improve efficiency of scheduling in CCS. My own instructional experience in the unit and a timely letter from the school at Sargodha helped me in getting started.

I would like to thank Squadron Leader Khalid P. Marwat for providing help in starting this project and Lt Col John Valusek for his help and advice. Lt Col Valusek's insights and guidance were instrumental in completion of this project.

I would also like to thank my wife, Roohi, and children, Hasan, Maheen and Amna for their support during my stay at AFIT.

Table of Contents

	Page
Preface	ii
List of Figures	v
List of Tables	vi
Abstract	vii
 I. Background	 1
Student Flying	1
Student Pilot Scheduling	2
Instructor Allocation	3
Student Controller Scheduling	3
Instructor Allocation	4
Current Methods	4
Additional Requirements	5
Schedule Alteration	5
Lead Mission Distribution	6
Peculiarities	6
Present Status	7
PC Based Scheduling Effort	8
Scheduling Requirements	8
School Scheduling	9
Maintenance Scheduling	10
Vehicle Routing and Machine Scheduling	10
DSS	12
Objective	14
Sub-Objectives	15
 II Methodology	 16
Components of DSS	17
Data Component	17
Model Component	18
Dialog Component	18
Phases of Problem Solving	18
Building a DSS	19
R.O.M.C. Approach	19
DSS Development	20
Adaptive Design	21
Kernel Selection	22
Concept Mapping	22
Storyboarding and Feature Chart	22
Hook Book	23
Evaluation	24
Summary	24
Methodology	25
Requirements Determination	25
Approach Selection	25
 III Specific DSS Development	 28

Phase-I	28
Why DSS Applies	28
Phase-II	29
Concept Map	29
Feature Chart	30
Storyboarding	32
Evaluation	33
Kernel Selection	33
Database Components	34
Model Component	35
Software Selection	36
First Iteration	36
Second Iteration	37
IV Resulting System	39
Data	39
Syllabus/Availability	40
Mutual Flying	40
Instructors	41
Availability Record	41
Formations	41
Dialog	42
Main Menu	43
Make Schedule	45
Altering Availability	45
Flying Schedule	47
Menu Hierarchies	48
V Conclusions and Recommendations	51
Specific DSS Conclusions	52
Model	53
Efficiency Verses Effectiveness	53
Recommendations	54
Adaptive Design Methodology	55
Software Selection	56
Users and Implementation	57
Recommendations	58
Appendix A: Storyboards	A-1
Appendix B: Users Manual	B-1
Bibliography	BIB-1
Vita	V-1

List of Figures

Figure	Page
2-1 DSS Development Process	24
3-1 Concept Map	31
3-2 Feature Chart	32
4-1 Database Relations	42
4-2 Planned Main Menu	44
4-3 Actual Main Menu	44
4-4 Planned Make Schedule Menu	46
4-5 Actual Make Schedule Menu	46
4-6 Planned Non-Availability Menu	47
4-7 Actual Non-Availability Menu	48
4-8 Planned Menu Hierarchy	49
4-9 Actual Menu Hierarchy	49

List of Tables

Table	Page
1-1 Segment of a Sample Schedule	2
4-1 Student Syllabus/Availability Database	40

Abstract

The object of this research was to develop a decision support system (DSS) for the scheduler in Combat Commanders School (CCS) of the Pakistan Air Force. The purpose of a DSS is to support the cognitive process of judgment and choice. Scheduling in CCS is a data intensive activity. Currently the scheduler accomplishes his task using grease boards and manual data tracking tables. Manipulation and cross-referencing of data for scheduling is a time consuming and complex process.

This DSS was built using the adaptive design approach. Ideally the iterative process of building needs the involvement of users for evaluation and re-definition of requirements. However, even if the users are not available, the adaptive design approach enables a DSS to stay open ended for incorporating future changes.

This DSS automates the data handling and enables a scheduler to build the scheduling shell from generic formation shells provided by the system. The DSS provides a screen format for simultaneous cross-referencing of data and scheduling. This DSS is likely to save a scheduler's time and help him in improving the efficiency and effectiveness of the process.

I. BACKGROUND

Combat Commander's School (CCS) is the flight leadership training institution of the Pakistan Air Force. The school simultaneously conducts courses on Mirage, F-6, and F-16 aircraft. Six or nine pilots from each weapon system are inducted in a course, which implies that either 18 or 27 student pilots (S.P.) get to fly together for the duration of the course. In addition to the pilots, 9 air defense controllers are also inducted in the course. The first month of the course is dedicated to academics. The remainder of the course period is used for flying.

STUDENT FLYING

The course is graded and a set syllabus is followed. The syllabus is divided into thirteen air to air and surface attack phases. The syllabus for the three type of aircraft differs from one another in some of the phases. This difference is created to balance the effects of aircraft performance on student performance. Students are exposed to dissimilar threat environments during most of the phases of flying. For example during the air combat phase a student will always be opposed by another student flying a dissimilar aircraft, and during the surface attack phase, dissimilar aircraft act as interceptors. In some of the phases a "mixed bag" formation (a single formation comprised of two type of

aircraft) is flown against the third type of aircraft. A typical flying schedule of one mission which depicts these aspects is shown in Table 1.1.

1.	FLT.LT	PILOT 1	LEAD	} Airfield Strike (MIRAGE)
	FLT.LT	PILOT 2	NO:2	
	FLT.LT	PILOT 3	NO:3	
	FLT.LT	PILOT 4	NO:4	
		and		
	FLT.LT	PILOT 5	NO:5	} Escort(F16)
	SQN.LDR	PILOT 6	NO:6	
		Versus		
2.	FLT.LT	PILOT X	LEAD	} CAP(F6)
	FLT.LT	PILOT Y	NO:2	
		S.C		
		I.C		

Table 1-1 Segment of a sample schedule.

In this representative schedule of one mission, four Mirages team up with 2 F16's for a strike. Two F6's from a separate formation act as adversaries of the first formation. This representation highlights only one segment of the airfield strike phase. To complete the phase each student is required to fly in all formation slots. Therefore, in subsequent missions the role of ESCORTS may be taken over by Mirage's and so on. In some other less integrated phases only two types of aircraft are scheduled for one mission. The number of missions to fly during a particular phase of flying differ from phase to phase.

STUDENT PILOT SCHEDULING

Student pilots lead the missions, with instructor pilots

(I.P.) flying in the NO:2 position. The remainder of the formation slots are filled by other students. The student syllabus specifies all the missions that a student is required to fly during the course. For example, in the Airfield Strike phase, an F16 pilot is required to lead a four ship strike. He also flies in the NO:3 and NO:4 slot of a strike, led by some other student. In addition to this he flies two escort leads (NO:5 of a strike), in support of F6 and Mirage strikes. Besides meeting the syllabus requirements, the scheduler pairs up the adversary formations such that each student gets to lead an equal number of missions against all other students. This removes the biases caused by an adversary student's performance.

INSTRUCTOR ALLOCATION The scheduler has to ensure that a student gets to fly with all the instructors equally. This is done to insure fairness in instructor grading. Since all the instructors are given equal amount of flying, monthly flying and availability also play a part in I.P. allocation.

STUDENT CONTROLLER SCHEDULING

Student controllers (S.C.) are involved in each phase of flying. One student controller is scheduled with a formation according to a controller syllabus. For example in the case of airfield strike an S.C. will control the F-6 and Mirage CAP formations. During surface attack phases only one formation gets a controller whereas during most of the air

combat phases both the sides are helped by controllers. The scheduler must insure that S.C.'s control the S.P.'s for an equal number of missions. This is done to remove grading biases caused by S.P./S.C. performance.

INSTRUCTOR ALLOCATION An instructor controller (I.C.) is scheduled with each formation for supervising/grading the S.C. Here again I.C.'s are rotated to balance the grading biases.

CURRENT METHODS:

Scheduling of pilots and controllers in CCS is presently done by one of the instructors, who handles this job in addition to his normal flying duties. Usually an experienced pilot who understands the school operations is assigned this job. A lot is demanded from students in terms of briefing, flying and debriefing. Therefore, to give sufficient preparation time, the flying schedule is issued two days in advance. In one of the phases this warning period increases to 3 days. Presently the scheduler uses a number of charts and boards to ensure syllabus adherence, I.P./S.P. matching, and deconfliction. The scheduler requires information about the following, before making the schedule.

1. Tomorrow's schedule.
2. Today's unaccomplished missions.
3. Remaining course syllabus for each student.

4. Aircraft availability.
5. Pilot / controller availability.
6. Student priority.
7. Instructor priority.
8. Availability of areas/range and their times.

Usually the scheduler starts by building formation shells for each type of aircraft. This depends upon availability of aircraft, current phase of flying and availability and timing of areas. These shells are filled by students based on their remaining syllabus. Then instructors are assigned to each formation to fill the NO.2 slot. The next step in the scheduling process is the allocation of student controllers and instructor controllers. Once the flying schedule "shell" is complete, Mobile Officer duties for the students, and Flying Supervisor duties for the instructors are scheduled. In addition to normal course flying, functional check flights (F.C.F.) and other ad-hoc commitments such as non-course flying or meetings are also scheduled. The accomplishment of ground duties and ad-hoc commitments takes precedence over flying. However, the flying schedule is usually made first because pilots always outnumber the available aircraft at any one time (i.e, either a non flying or physically unfit pilot is available for this job). The only exception is when sufficient number of pilots are not available. '

ADDITIONAL REQUIREMENTS

Besides making a normal schedule there are some factors that put extra constraints on the process of scheduling.

SCHEDULE ALTERATION The scheduling process takes a new dimension when one of the students is sick once the schedule has been issued. Under these circumstances a non flying S.P. usually fills the vacant slot. However, if no extra S.P. is available, or the available S.P. has already completed the available mission, shuffling is required to fill the spot.

LEAD MISSION DISTRIBUTION Another aspect which the scheduler must address is the distribution of extra lead missions. When one or more students are removed from the course (because of poor performance), extra flying by the remaining students is required to meet the demand of other classes. Since this extra flying is also graded, the scheduler has to ensure that it is divided equally among all remaining students of that particular class. Lead missions hold a special importance because failing can ruin a student's grade. Failure in an uncalled-for extra lead mission has even resulted in suspension if the student concerned is a border-line case.

PECULIARITIES:

The scheduling in CCS differs from a normal fighter squadron because it involves co-ordinated scheduling of pilots from three classes (squadrons). The syllabus followed also differs from the training syllabus of a squadron. The cyclic training syllabus of a normal fighter squadron lays down

general guidelines for flying and does not specify each and every mission in detail. The trivial details like formation breakdown and formation matching are not given any importance because no assessment is done. For example, if it requires that five 2 Vs 2 dissimilar sorties be flown by each pilot during the training cycle, the scheduler will put the section leaders in the lead slot and fill the remaining slots with wing men. In case no wing man is available, a section leader can fill the NO:2 slot. The overall aim of the scheduler is to give each pilot the required number of missions while considering individual flying qualifications. This process requires consultation from both the syllabus and flying hours databases. However in CCS the scheduler is required to schedule three different classes of students to fly against each other while considering the S.P./S.P. matching, S.P./I.P. matching, S.P./S.C. matching and S.C./I.C. matching constraints.

PRESENT STATUS

Currently the CCS scheduler relies upon manually maintained charts and boards and spends a large amount of his working time making the schedule. As the scheduler holds this job in addition to his flying duties, he is always short of time. Shortage of time and manual processing results in overlooking one constraint or another by the scheduler. The course critiques invariably point toward this act of mismanagement. The lack of time and requirement to consult an

extensive set of data affect the efficiency of the process. Therefore, a tool is needed to support and improve the efficiency of this process that now must be done manually. The scheduler could possibly benefit from automation support to make the schedules while meeting all the requirements. There is a computer available in the school, but it is used only for word processing. All the unit records (including flying records) are maintained manually. Since scheduling is a data intensive activity, the automation support which the scheduler needs can also be a first step toward automation of the unit records. This system should be easy to use and must offer a time saving benefit. It must provide sufficient user control in accommodating his assigned priorities, while at the same time be capable of providing automation support for filling the formation slots.

PC BASED SCHEDULING EFFORT

Scheduling is defined as the process of loading, sequencing, dispatching, controlling and updating activities (Dervitsiotis:595). Loading involves the process of assigning workers to jobs (Dervitsiotis:597). Sequencing is: "...the time ordering of jobs through one or more processing centers according to some criteria of performance" (Dervitsiotis:607). Dispatching is the actual performance of the task. Controlling involves monitoring the performance status and making updates whenever required (Dervitsiotis:595). Analytical techniques have been used in the past to develop

scheduling systems. However, due to a number of reasons which will be discussed in the following pages, the optimization techniques have not been used by users.

SCHEDULING REQUIREMENTS: There are a number of reasons which make analytical techniques unsuitable for pilot scheduling. As mentioned earlier, pilot schedules are required to be changed quite frequently before actual flying. This happens because of unexpected sickness, meetings etc. A program that optimizes the output will completely change the schedule when only a minor adjustment is required. Occasionally a lot of changes are necessary to accommodate a minor adjustment; however, the scheduler always attempts to keep the changes to a minimum. This helps the pilots in their preparation for forthcoming missions (Doug). The second reason is the lack of flexibility that an optimization program offers. Sometimes it is necessary to change the priorities according to which pilots are scheduled. To modify the program to accommodate the new priorities will require the services of an expert (Pannell) who is usually not available in a normal squadron. Thirdly, a scheduler is more than often interested in a schedule which fulfills the training requirements of the pilots. The scheduler is concerned with meeting the requirement. If a serviceable aircraft is not flown on a particular day, the scheduler is not much concerned about it as long as the day's training requirements have been met. The scheduler does not care about optimizing the output.

He is concerned about making A SCHEDULE. Commercially available software packages lend themselves well for this particular problem. The work done by different researchers using PC based techniques will be reviewed in the following pages.

SCHOOL SCHEDULING: The school scheduling problem is similar to pilot scheduling in many respects. Honour (Honour) in his thesis formulated the pilot scheduling problem as a classroom scheduling problem. He developed an algorithm to facilitate scheduling in a navy training squadron. The algorithm solves a cost matrix to assign instructors to aircraft. The assignment of students is done by a separate heuristic. Honour's algorithm does not meet the requirements of flight scheduler because it only assigns instructors/students to aircraft. The system does not consider many other factors like the student syllabus, special requirements of the squadron, assignment of ground jobs, scheduling of classes, etc.

MAINTENANCE SCHEDULING: Typically maintenance planning involves the scheduling of crew and airplanes. The objective is to minimize cost while ensuring that the required inspections and maintenance actions are performed according to the tech orders (or schedule). Holst (Holst:735) generated an interactive combined scheduling and maintenance DSS for SAS airlines. The model component of this DSS is an integer

programming based optimization algorithm. Pilot scheduling and maintenance scheduling are similar as far as structure and allocation of resources are concerned. The structure is provided in the flying syllabus in the case of flying scheduling, and by tech order inspections in the case of maintenance scheduling. The difference that separates these two types of scheduling is relative inflexibility of maintenance systems (Trapp:12). Pilot scheduling needs a flexible system which can take care of last minute changes in the schedule.

VEHICLE ROUTING AND MACHINE SCHEDULING: The scheduling of vehicles involves finding minimum cost routes originating and terminating at a point and serving customers with known demands at specified times. Each customer must be serviced only once. All customers must be assigned to vehicles without exceeding their capacity (Bodin:63). This is similar to machine shop scheduling where jobs (customers) are assigned to machines (vehicles), the goal being optimal assignment for timely completion of jobs. In terms of pilot scheduling this can be viewed as pilots being assigned to aircraft or other duties.

Huelskemp (Huelskemp) treated the pilot scheduling problem in a similar fashion. Huelskemp developed a program for schedulers at an undergraduate pilot training (UPT) squadron. The problem has been formulated as a vehicle

assignment problem, and solved in two parts. In part-1, which the author calls level-1, network paths depicting instructors are found using Dilworth's decomposition algorithm. In level-2, students are paired with their assigned instructors just as the vehicles are assigned in an assignment problem. Costs are assigned to indicate desirability of a particular pilot's schedule. This model falls short of a scheduler's requirements because it does not facilitate assignment of ground duties. The model also does not consider the training syllabus while making a schedule. In another similar work Mathews (Mathews) developed a heuristic to assist the scheduler in a tactical fighter squadron. Mathews formulated the problem as a transportation problem. A primal network simplex method was then used to assign pilots to aircraft or other ground jobs. His assignment problem maximized the benefit achieved when a pilot is assigned to a job (A pilot's assignment to a job accrues certain benefits which differ from job to job). Pilot suitability and the number of missions available are taken care of in the constraints of the problem. This model, like the previous model, did not consider tracking of the training syllabus.

Roege (Roege) developed a mathematical programming model which deals with flight scheduling in an F-15 squadron. Roege formulated it as an assignment problem, which assigned pilots to specific duties at specified cost. This model generates a flying schedule which ensures equal flying for the pilots.

Scheduling of ground activities, tracking of the training syllabus, and pairing of pilots were beyond the scope of this study.

Pannell (Pannell) investigated the feasibility of using linear programming techniques for scheduling pilots. His research did not consider all the variables that complicate the scheduling problem. In Pannell's model, priorities are assigned to the pilots, and available slots are filled according to this priority. The remaining slots are filled by lower priority pilots. This model did not consider scheduling of other ground duties.

DSS

Optimization techniques work best when applied to solve structured problems. However, semi-structured and unstructured decisions can't be optimized by using linear programming or other such techniques in a reliable and predictable way (Keen & Morton:11). These types of decisions involve matters requiring judgment and choice. Creating or visualizing the alternatives and picking the right solution is best done by a human being working in the loop.

The purpose of a DSS is to establish an integrated framework between the problem, the machine, and the decision maker. The DSS couples the speed and thoroughness of automation with the insight of human experience and, where appropriate, a proper blend of quantitative support (Davis:5)

Capt James H. Jeter (Jeter) in his thesis "On Site Adaptive Design Of A Decision Support System for Fighter Squadron

Scheduling" developed software to make a monthly schedule in a reserve squadron. The thesis did not address daily scheduling as the users were reluctant to abandon their current, manual grease board methods. This DSS was developed using ENABLE integrated software.

Captain Sahli & Captain Shacklett (Sahli & Schacklett) in their thesis " A Prototype Microcomputer Decision Support System for Air Crew Scheduling " built a microcomputer based DSS for a multi-crew special operation squadron. Their aim was to evaluate the feasibility of a microcomputer based DSS, not its effectiveness or efficiency. The DSS was developed using a commercial package of BASIC subroutines. Management of data was done by retrieving and manipulating records from a data record file using one of the BASIC subroutines.

In a similar thesis Captain Kopf (Kopf) developed a scheduling system for RC-135 aircraft. This DSS was also built using non-analytical techniques. The major emphasis of this work was on the database component of DSS. The author writes: "The database component is perhaps the most important component in the scheduling decision aid.....air crew scheduling process is data intensive." A model component of the DSS was developed using the spreadsheet capability of ENABLE.

Captain Paul E. Trapp & Captain Jeffrey W. Grechanik (Trapp:) worked on flight scheduling in their thesis "Design

Evolution of a Fighter Training Scheduling Decision Support System". They built a DSS which provides the required interactive support to the scheduler. Their program was coded using a spreadsheet. This thesis produced software which can sort the availability of pilots, read the wing data file (containing flying area and area times) to build a scheduling shell and maintain a deconfliction database. The authors proposed the following continuations to their research:

1. Addition of student syllabus database.
2. Addition of academics, duties/meetings, simulator, etc to the deconfliction database.
3. Matching of I.P. and S.P. according to syllabus requirements.

This DSS, like the others mentioned above, fulfills the requirements of a particular user. However it can be expanded to accommodate the requirements of any fighter squadron.

OBJECTIVE:

It was the purpose of this research to investigate the scheduling process in general, and the approach taken by Trapp & Grechanik in particular. If their work was found suitable for expansion to meet the requirements, this study would have expanded upon their work to incorporate the additional requirements of CCS. Since their work could not be adapted, this study built a microcomputer based scheduling decision support system for CCS.

SUB OBJECTIVES:

1. Use the adaptive design methodology currently being researched at AFIT to evolve the DSS design.
2. Investigate the applicability of Trapp & Grechanik's approach to this problem.
3. Investigate the ability of software developed by the above mentioned authors to absorb the expected expansion.
4. If steps 2 and 3 had allowed further research, then their model would have been expanded to include:
 - a. The student syllabus database for actual scheduling by the system.
 - b. S.P. / I.P. matching algorithm.
 - c. S.C. allocation algorithm.
 - d. Formation matching algorithm.

Since the previous study could not be adapted, then:

- a. Suitable software was selected for accomplishment of the task.
- b. A kernel was identified to start the building process.
- c. The basic kernel model was expanded to include other requirements.

Chapter II discusses the Decision Support System concepts and components. The adaptive design approach to DSS development is described.

II. METHODOLOGY

The concept of DSS was first put forward by Michael S. Scott Morton. It is defined as : "...interactive computer based systems that help decision makers utilize data and models to solve unstructured problems" (Sprague and Carlson:4). It is merely a tool that supports the decision process. An important aspect of DSS is the role of the pervasive activity of judgment and choice. This activity may involve either value judgment or predictive judgment. Value judgment is hinged on decision maker's preferences, and predictive judgment is affected by decision maker's outlook (perception of information) and limits on information processing (Hogarth:3). As both these activities require human involvement, any supporting system must cater to this requirement. DSS is meant to "...support rather than replacement (automation) of human cognitive process" and it is "...an attempt to follow and facilitate the natural human process rather than to force fit it into a designer's notion of the best process" (Young:1). DSS is characterized as a system to:

1. Assist managers in their decision process in semi-structured tasks.
2. Support, rather than replace, managerial judgment.
3. Improve the effectiveness of decision making rather than its efficiency (Keen and Morton:1).

The results of predictive or value judgment are intimately linked with the limits on human information processing. These

limits are due to memory, lack of intuitive calculation capacity, and nature of human processing (Hogarth:5). DSS's consist of elements which fill this void.

COMPONENTS OF DSS

"Data", "Model" and "Dialogue" are the components of DSS. According to Sprague and Carlson, the integration of model and data components moves an information system from being an MIS to DSS (Sprague and Carlson:259). Dialog, the third component is an interface between the user and the machine required for handling data and models. "If any one of the three components is weak, this three-legged stool collapses"(Valusek)

DATA COMPONENT The data component supports the memory requirements of the user. "For a manager to find relevant information in a mountain of raw data can obviously be a non trivial task" (Keen and Morton:97). The type of data support provided to the user depends upon the sophistication of the DSS itself. An effective data management component must be able to filter, and recognize the patterns for effective retrieval. In addition to this, the ability to perform simple calculations, comparisons and projections will mimic the normal habit patterns of managers (Keen and Morton:97). How much data to collect, maintain, and display to the user, entirely depends upon the type of decision involved.

Only by focusing on the decision first and then defining the information required to support it, is it possible to

see which data are worth collecting and where the collection and maintenance process should take place (Keen and Morton:85).

MODEL COMPONENT

The model component of DSS is meant to:" provide the managers with answers they can and will act on" (Keen and Morton:97). The usefulness of answers generated is the deciding factor for selection of modeling technique. The technique selected for modeling depends entirely upon the nature of problem. It may be mathematically sophisticated or be very simple like a basic spreadsheet model.

DIALOGUE COMPONENT

This is the single most important component of DSS. It forms the interface between the user and the system. This holds a special importance because it alone determines the usefulness of the DSS. Dialog alone determines the usability of a DSS because: "In fact from the DSS users point of view, the Dialog is the system" (Sprague and Carlson:29).

...incompatibility between the decisionmaker's problem solving habits, strategies, and abilities and the implicit style of the system will generally result in it not being used" (Keen and Morton:73)

The dialog component must satisfy the objective demands of the task and fit the cognitive style of the user (Robey and Taggart:281). In other words "the system should mesh with the cognitive structures of the users" (Keen and Morton:73).

PHASES OF PROBLEM SOLVING

Simon defines the three phases of problem solving as "Intelligence", "Design", and "Choice". Intelligence is

recognizing the problem from its environment. That is pinpointing and enumerating the decision. The gathering of data is also done in this phase. The design phase involves the creation, development and analysis of alternatives. Creativity is the key to successful completion of this phase. Imagination, and open mindedness is beneficial for this activity (Hogarth:110). Choice is the selection and implementation of the most suitable alternative (Simon:3). These three phases are best explained by Keen and Morton as: "What is the problem? What are the alternatives? Which is best?" (Keen and Morton:95).

The DSS must provide support in all three phases of decision making. Data handling and manipulation capability support the decision identification and alternative generation, whereas ability to examine various options helps the choice phase.

BUILDING A DSS

For a DSS to be successful, the capabilities of the DSS and requirements of the decision maker must match. The decision makers usually find it hard to describe the decision support required. Sprague and Carlson identified an approach to identify requirements in each of the three capability areas of DSS.

R.O.M.C. APPROACH

The letters R, O, M and C stand for Representations,

Operations, Memory aids, and Control mechanisms respectively. The objective of the R.O.M.C. approach is to bridge the gap between the requirements of the Decision Maker and capabilities of the DSS. " The most important characteristic of the ROMC approach is that it is a process independent approach for identifying the necessary capabilities of a specific DSS" (Sprague and Carlson: 101). These terms are defined as:

Representations: Way the system will present the decision situation to the user.

Operations Way the user will perform operations to analyze and manipulate those representations.

Memory Aids: To assist the user in linking representations and operations.

Control Mechanisms: The control mechanisms that the user employs to handle the entire system (Sprague and Carlson:96).

The ROMC are developed in terms of Simon's Intelligence, Design, and Choice activities and provides a framework for identifying the required characteristics and capabilities of a particular DSS.

DSS DEVELOPMENT

Developing a DSS is a phased process. Young outlines a three phase approach to its development. Phase one is preliminary assessment, phase two is prototyping, and phase

three is iterative usage and further development of the system (Young:171). Phase-I involves preliminary assessment of the problem or requirements analysis. This is done to find the applicability of the DSS approach and determine the user requirements. If the DSS approach is found usable then the next step in system development is taken. According to Young, the next phase is a rough working model of the system for analysis and on-site improvement. Another substitute for this phase is the Adaptive Design approach. Adaptive design refers to design and development of the system at the user's facility as opposed to rapid prototyping which is done at a builder's facility (Valusek:105).

ADAPTIVE DESIGN

During a conventional design approach, requirements are frozen at one time or the other and the control is handed over to the builder. The scope and shape of the system which has not yet been built is decided with this step. However due to a number of reasons the users are unable to provide functional definitions or are unwilling to do so. This may be because the user does not know what he/she wants (Keen:15). Therefore an approach which can cater to changing or increasing requirements will work better than the approach in which the user has no say after the design "freezing" phase. Another reason which supports this approach is the fact that users usually learn from the DSS and want to change their requirements (Keen:15). This approach makes the whole design

process shorter but it turns it into an iterative process which ends only when the user is completely satisfied. The adaptive design of the system starts from a small but central part of the problem (kernel) and then grows with each successive iteration. The user is never left out of the development loop. New user requirements that emerge as a result of the learning process or that emerge otherwise can be assimilated by the growing system.

KERNEL SELECTION

Identification of the kernel, the critical component of the problem, is critical to the adaptive design approach. "Kernel selection is critical to the development of an effective base system that can continue to evolve" (Jeter:21). Finding the kernel from an ill defined problem space needs the help of several techniques like concept maps, storyboarding and feature charts (Valusek:107).

CONCEPT MAPPING A concept map is a graphical representation of a concept drawn using key ideas and their relationships. Its purpose is to communicate knowledge and understanding (Novak and Gowin:15). The mapping is done time and again to draw knowledge from all corners of memory. The resulting map is reordered so that a stranger can make sense out of it. Initial concept mapping should be done on an individual basis (Valusek: 107). The concept map is a good way to communicate a stranger about a problem without going

into the exercise of requirements statement. It also highlights the user's perspective about certain aspects and their relationships which cannot otherwise be fully captured using more structured techniques.

STORYBOARDING AND FEATURE CHART Once the problem boundaries have been identified graphically in a concept map, the next step is drawing the screen displays of the intended DSS. This drawing should be done by the user and does not consider the builder's limitations. The storyboard tool depicts the wishes of an intended user. The ROMC approach is implemented as the guideline during this process. Subsequently a feature chart is drawn to graphically represent the storyboards for better understanding of connectivity.

The purpose of storyboards is to represent all of the important functions that the system will perform as well as the output that it will generate (Andriole et al, 1987:5-6)

The selection of the kernel is made after viewing the concept map and the storyboards. Once the building process starts any new emerging thoughts are stored for future incorporation.

HOOK BOOK A "Hook book" is maintained to keep a record of any new ideas or thoughts which are presently not part of the system but are thought to be necessary for system improvement (Valusek: 109). These entries are later added to the system requirements during the subsequent iterative cycle

of adaptive design. The entries in the Hook Book are dated along with the circumstances leading to that particular entry. The date and circumstances help in recalling the details of the entry at a later time. These entries can be subsequently categorized under different DSS headings in a chronological order.

EVALUATION

Evaluation of a DSS starts with the conception of DSS and continues throughout the life of the DSS. Evaluation is done during the conception stage to assess the requirements that have been determined. A critical analysis of requirements will invariably highlight some of the neglected or over-emphasized aspects. Evaluation is done by applying Sprague and Carlson's four "P's" approach. These four possible measures are:

Productivity measures are used to evaluate the impact of DSS on decisions.

Process measures are used to evaluate the impact of the DSS on decision making.

Perception measures are used to evaluate the impact of the DSS on decision maker.

Product measures are used to evaluate technical merits of the DSS (Sprague and Carlson :159).

SUMMARY

The three phases of DSS development outlined by Young (Fig 2.1) provide an overall structure for the designer. The designer can use other tools (like adaptive design) and techniques within this structure.

DSS DEVELOPMENT PROCESS

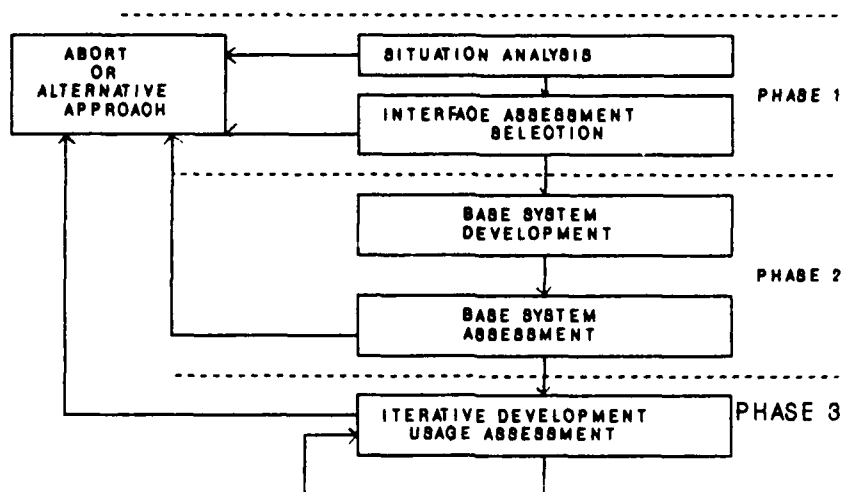


Fig 2.1 DSS Development Process

In the first phase of DSS development requirements analysis(situation analysis) is carried out and feasibility or cost effectiveness of DSS (Interface assessment/selection) is determined. The second phase corresponds to prototyping(base system development) and assessment. The third stage involves further iterative development and assessment.

METHODOLOGY:

The DSS approach as documented in the preceeding pages was used for development of the proposed Scheduling DSS. The exact methodology adopted was as follows.

REQUIREMENTS DETERMINATION

The whole cycle of the scheduling process was

documented and the concept map was built from personal experience. Help from a few experts, currently stationed in the United States, was provided, to refine this concept map. Based on this initial cut, the whole process of scheduling was analyzed (i.e., traced the process of data input and output as experienced by a user). Based on this information, the storyboarding process (i.e., building the screen displays that will be essential to support the scheduling process) was completed. The storyboards represent user's requirements under ideal information flow and processing conditions.

APPROACH SELECTION

The next step involved the selection of an approach for building the system. Two possible choices were available. One using Trapp and Grechanik model as a starting point and then building upon it to meet the requirements. The other choice was to start from the beginning. Trapp & Grechanick's model was studied to ascertain its ability to absorb the increased requirements. Their model was studied in the light of the proposed system requirements to ascertain its feasibility for adoption. The results of this study were as follows.

1. The Model was designed to accept input from a wing data file to construct the basic scheduling shell for each day, whereas the proposed system requirements envisaged user defined inputs for construction of this shell. Their major effort was spent on translating the wing file into a usable

schedule shell.

2. The requirements analysis showed the data intensive nature of this specific DSS. Trapp & Grechanik's model did not support the increased data requirements. Their DSS was built using only the spreadsheet component of ENABLETM because of the inability of the software to interact freely between the spreadsheet and database. The spreadsheet has a limited data processing capability but it is inefficient and requires a large disk space and computer memory to store and use this data. The new version of ENABLE is believed to have improved capabilities but it was decided not to learn the software and understand the previously written macros only to find that the approach may not work.

The inability of ENABLETM to support free interaction between the spreadsheet and database and the limited time to indulge in experimentation left the author with no choice but to abandon the Trapp & Grechanik approach. Secondly it was felt that the effort spent in understanding their model would far outweigh the starting advantage (i.e., maintenance of an availability list and a deconfliction chart) that it would give. Therefore the second approach to building the system was adopted.

An adaptive design approach was taken to build this system. The building process was started by selecting a kernel. A base system was built around it, then additional requirements were added to make the model grow. A "hook book"

to record all new ideas and thoughts which were not included in the model was maintained and consolidated into a "road map" for the system to progress.

Chapter III describes the application of the adaptive design approach.

III SPECIFIC DSS DEVELOPMENT

The actual development of DSS was carried out using the three phase development process as outlined in the second chapter.

PHASE-I

The first phase in the development of DSS is "requirements determination". Ideally this process is best captured when the would-be user is available for frequent consultations and feedback. The unit for which this DSS is intended is located on the other side of the globe in Pakistan. The distance involved restricted the involvement of a user. However, the author's previous scheduling experience and a detailed letter from the present scheduler did fill this void to a great extent. It should be noted that no process improvement is required. The present method of scheduling is working fairly well. However, it is the efficiency of the process which needs to be improved.

WHY DSS APPLIES: Scheduling in CSS is a process which is data intensive. It requires a lot of data manipulation for answering the key question:

"Who should fly in which formation slot".
The scheduler needs a data management system and a man-machine interface (dialog) which not only facilitates retrieval of

this data for viewing but also supports a screen format for simultaneous acceptance of the user inputs required for scheduling. The scheduler selects the formation leaders, instructors, and student controllers based on requirements mentioned in the first chapter. The remaining formation slots usually do not warrant any stringent selection process (as only syllabus requirements need to be met). Therefore, a component (model) which can provide automatic filling of the scheduling shell when the scheduler so desires is also needed. Manipulation can be done by an MIS but to support the second and third parts of this process a DSS is required.

PHASE-II

The purpose of this phase was to produce a working model of the DSS which could then be developed into a full scale DSS. The adaptive design approach was taken to accomplish this task. The design process started by selecting a suitable kernel around which the system could be built. The kernel selection techniques of concept map, storyboarding and feature chart were used.

CONCEPT MAP

This exercise started with the listing of major tasks and activities of the process. All the aspects of scheduling process were identified and listed. The concepts were ranked in the order in which they would be performed. A list of these ordered actions is given below

1. Check completed flying and update the remaining syllabus database.
2. Check tomorrow's schedule and temporarily update the remaining syllabus database.
3. Update availability database of instructors and students.
4. Check availability and timing of areas/range.
5. Check student syllabus and make formation shells.
6. Fill the formation shells with students.
7. Check student/instructor record of flying and allocate instructors.
8. Do the same for controllers.
9. Check student vs student flying record and pair the formations.
10. Fill the remaining duty slots.(Mobile officer and SOF)
11. Ensure deconfliction throughout.

Finally these ordered concepts were linked together to form a rough concept map of the problem. Many further iterations helped in refining the map. This exercise further proved the data intensive nature of the process. This concept map was then reduced to depict only the major concepts to facilitate the selection of the kernel. Figure 3-1 shows the final concept map which includes the important aspects of the whole process.

FEATURE CHART

The process followed by the scheduler was then represented graphically to highlight the hierarchical

relationships. This graphical representation was required for the completion of the next phase of the kernel selection process. Fig 3.2 shows the feature chart derived from the concept map.

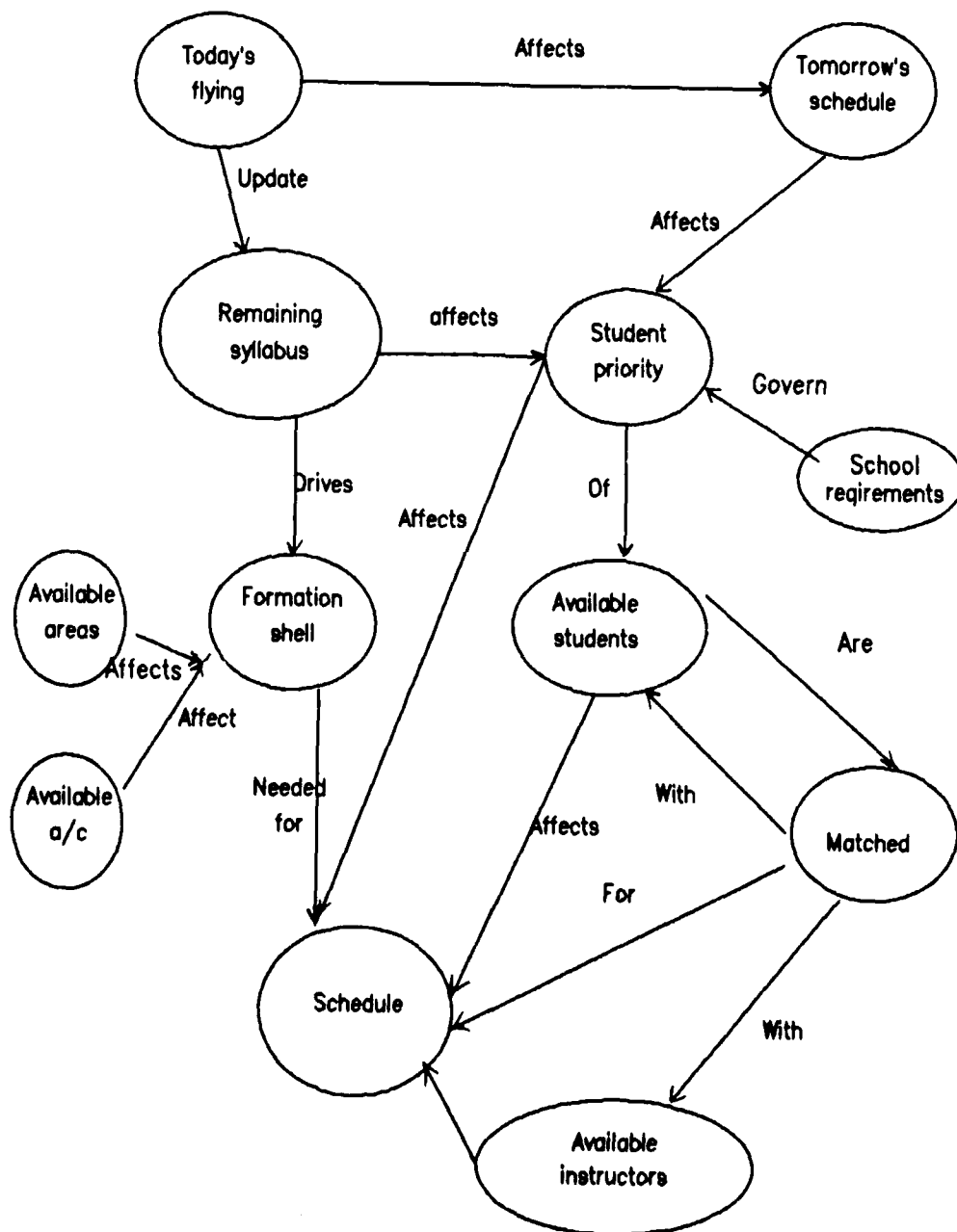


Fig 3.1 Concept Map

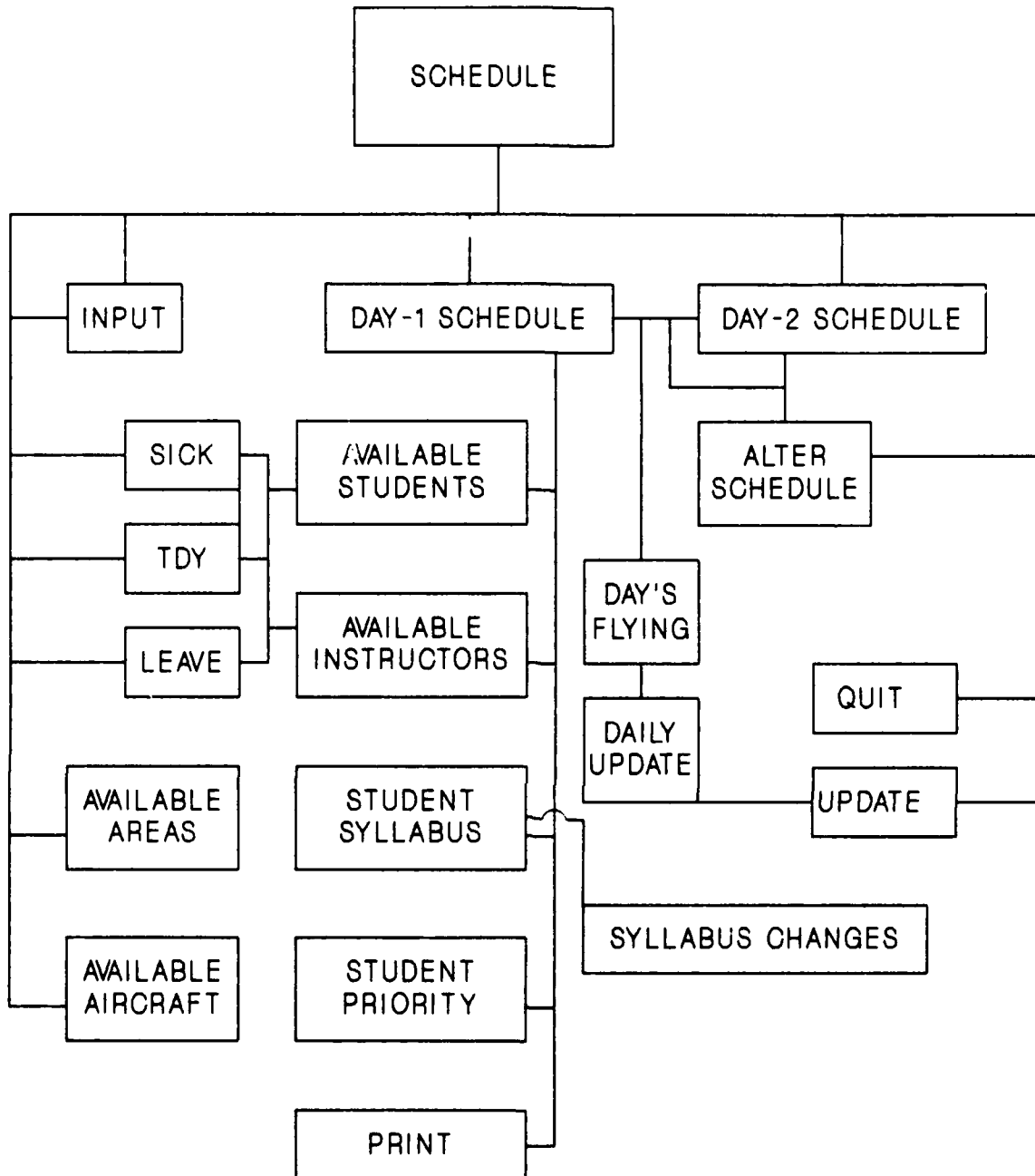


Figure 3.2 Feature Chart

STORYBOARDING

The next step was the drawing of storyboards. The feature chart was the key starting point in this direction. The sequence of the scheduling process was the other input

that was used to build the storyboards. Sprague and Carlson's ROMC approach was used as a check list to capture the process. User requirements in terms of process flow were incorporated. User knowledge of the computer was considered and operations were simplified. On-line help for each screen was provided at the push of a key. Ideally the storyboards should have been drawn by the user and arranged and grouped for screen display by the builder so that the user's desires and wishes and the decision process are fully captured. The objective is to capture ideal scheduling screen displays properly linked without regard to technological restrictions. Since it was not possible to elicit users help, this exercise was done by the author alone. All effort was made to remove the biases of a builder's perspective.

EVALUATION:

A preliminary evaluation revealed many drawbacks in the storyboards. Original storyboards did not allow enough user control. The level of automation left the scheduler out of the scheduling loop. This was thought to generate hostility due to lack of control over the process. Another drawback noted was lack of memory aids on the screens. These aspects were later added to the modified storyboards. The storyboards are attached as Appendix A to this thesis.

KERNEL SELECTION:

There are four important aspects of the problem which a

scheduler will pay the maximum attention to. They are:

1. Following the syllabus.
2. Allocation of instructors to students.
3. Matching S.P.'s. with S.P.'s. while pairing up the formations.
4. Allocation of S.C.'s. to S.P.'s.

Following the syllabus is the most important aspect and the scheduler will give it the highest priority. Although the syllabus is in a database which is consulted during scheduling, it drives certain decisions. While making the schedule the scheduler has to consider the remaining days and remaining sorties for each pilot. The scheduling should be done such that the following day's formation can be formed without anybody having to fly extra missions. The scheduler must stick to the syllabus while considering the formation matching and instructor allocation requirements. The second and third aspects are assigned lower priority. Secondly building the system around the syllabus can facilitate later changes to the system. Therefore syllabus tracking was taken as the key kernel.

DATABASE COMPONENTS:

Those aspects of the DSS which require databases to be built were pinpointed. It was found that this DSS requires many databases. The number of databases is high because of the peculiar nature of scheduling. As the syllabus is different for different type of aircraft and the students are

scheduled against each other, the number of databases required is therefore quite large. The databases which were identified for the system were:

1. Four syllabus and availability databases for each class of students. A separate database for each class is required because of differences in their syllabi.
2. Instructor's availability and flying hours database.
3. A mutual database to keep track of flying in terms of:
 - a) who flies against whom as far as S.P.'s are concerned.
 - b) who flies with whom as far as S.P.'s & instructors and S.P. & S.C.'s are concerned.

MODEL COMPONENT:

The process of scheduling is a semi-structured problem. The scheduler has to meet the syllabus and mutual flying requirements under ideal circumstances. However, the circumstances are not always ideal. For example, a student may not feel comfortable flying with a particular instructor or another student scheduled in his formation. Similarly at another instance the squadron commander might like to fly with a particular student even though he may not be eligible to fly with him as far as the entries in the mutual database are concerned. In the absence of these and many other such impediments, the process becomes fairly structured. If the scheduler has met all the requirements and scheduled all

leaders and other such pilots who need special care, the remaining task becomes fairly structured. This remaining structured component which usually involves scheduling of NO:3 and NO:4 could be handled by an automatic model component with greater efficiency.

SOFTWARE SELECTION

Software selection was done keeping in mind the requirement for extensive data management, the required man-machine interface and the ability to accommodate a model component. The first two requirements can be met by any integrated software package. However it was the model component which had something to do as far as selection of software was concerned. It was felt that a rule based model component would do the work efficiently. Therefore the VP series of integrated software by Paperback SoftwareTM was selected for building this DSS. This series is comprised of VP-info (database), VP-planner (spreadsheet) and VP-expert (a rule based expert system shell).

FIRST ITERATION

The prototyping of the actual DSS started with the construction of all the databases and a system to manipulate them. This was done using VP-info. This step was found to be essential for starting the development of the DSS because of it's heavy reliance on data. Lack of any existing data management system in the intended user squadron made this task

even more important. This portion of the building process consumed the maximum time as it involved both learning the database software and building one subsequently. Till this time the plan was to keep VP-expert as the home operating environment and make calls to VP-info and VP-planner whenever required. The available version of VP-expert was not capable of handling the database call, however, it was found from the software company that their version 2.1 will perform the task. Based on their assurances, it was decided to continue with the development of the data management system. However, once version 2.1 was acquired and tested, it too was found to lack this ability. Though, theoretically the system is capable of handling database calls, it is severely handicapped by available computer memory. At this time it was decided to abandon the use of VP-info and rely totally on VP-expert and VP-planner. Before taking this new route it was decided to check the software compatibility/workability thoroughly. A little work in this direction highlighted the severe limitations and the problems that the software possessed. VP-expert was found to be capable of calling worksheets for operation but left only 28 K of memory for the user. It was felt that this limited memory might not be able to handle the system requirements. This prompted the author to abandon the use of VP-expert and rely only on the use of other two software.

A working system which initially operated from the VP-info environment for data management, and then switched to

VP-planner for actual use of data and receiving the inputs was built for initial testing. As there was no way to involve the intended users, the author's advisor was used as a "surrogate user" for iterative evaluation of the system. His evaluation pinpointed the need for maintaining past flying records of individual students and a requirement to graphically represent the daily schedule for easy perception by the viewer.

SECOND ITERATION

This iteration started with the inclusion of hook book entries in the system requirements. It was not possible to incorporate all the entries. Therefore only those entries which were found to be essential for the working system became requirements at this stage. The addition of graphical displays of the daily schedule and removal of other glitches from the system were the two basic tasks handled in this phase of building.

The development was terminated at the second iteration because of time constraints.

Chapter IV identifies the major elements of the Decision Support System that resulted from the adaptive design approach.

IV. RESULTING SYSTEM

The scheduling system provides decision support to the scheduler for daily flight scheduling at CCS. The system caters to the data browsing requirement of the scheduler and provides the interactive support for selection of pilots and controllers such that scheduling requirements are satisfied. The system was built using the VPTM series/integrated software package. VP-PlannerTM, the spreadsheet component of the package, was used to provide scheduling support to the user. Windows were used to enable the user to review multiple databases simultaneously. All important actions were supported by menus or imbedded macros through the use of function keys. VP-InfoTM, the database component, was used to handle the data requirements of the system. The use of Database software also provides increased capability to improve the system at a later date for data automation in the unit. The following sections of this chapter discuss the "data" and "Dialog" components of this DSS.

DATA

This DSS was found to be data intensive in nature. It was seen from this research and review of previous research in this area that a scheduler spends his maximum time on data maintenance. Therefore a major effort went into providing this needed support to the scheduler. The databases built for

the scheduler are many more than what were identified in Phase-I of the DSS development process (which are listed on page 34). Fourteen additional relations which store the basic formation shells for each phase of flying are built to aid making of a scheduling shell. These relations were required to save the user from typing the individual formation shells. Storage and manipulation of all the required data is handled by the data management component of the system. The data files and their contents are explained below.

1. Syllabus /Availability Four files are maintained to keep a record of the remaining syllabus for the four classes of students. The need for four different files arose because the syllabus for each class is different. Each sortie type of a syllabus occupies a unique column. Since the flying day is divided into three distinctive parts, three columns, which tell the user about the availability of pilots/controllers in each part of the planned schedule day are included. The data management component is capable of automatic updating of the remaining syllabus. A representative data table is shown in Table 4.1.

2. Mutual Flying This file contains the record of mutual flying. All the names are listed in the first horizontal row and vertical column. The intersection of names contain the number of sorties that one has flown with the other. This list is also automatically updated by the data management component.

NAME	FIRST	SECOND	THIRD	MSN1	MSN2MSNn
...
...
...
...
...

Table 4.1 Student syllabus/availability database.

3. Instructors This file shows the availability of the instructor pilots and instructor controllers on the date scheduled.

4. Availability Record This file is used to keep a record of forecasted non-availability. The relation was built such that the user can store the names of all pilots and controllers who are expected to be non-available on any future date. The date of non-availability is also stored. This differs from the syllabus/availability relation because it can store future information, whereas the syllabus/availability relation retrieves this information for the date of the schedule and displays it to the user.

5. Formations This is a set of fourteen database files each of which contain formation shells for one particular flying phase. These files could have been compressed into one big multidimensional database, but it was decided to keep them separate for ease of handling by the user. This separation

will be useful in the future if the structure of a particular flying phase changes and amendments to the system are necessary. These basic shells are used to construct the daily flying schedule shell before actually making the schedule.

A relational diagram of the data structure is shown in Fig 4.1.

DATA STRUCTURE

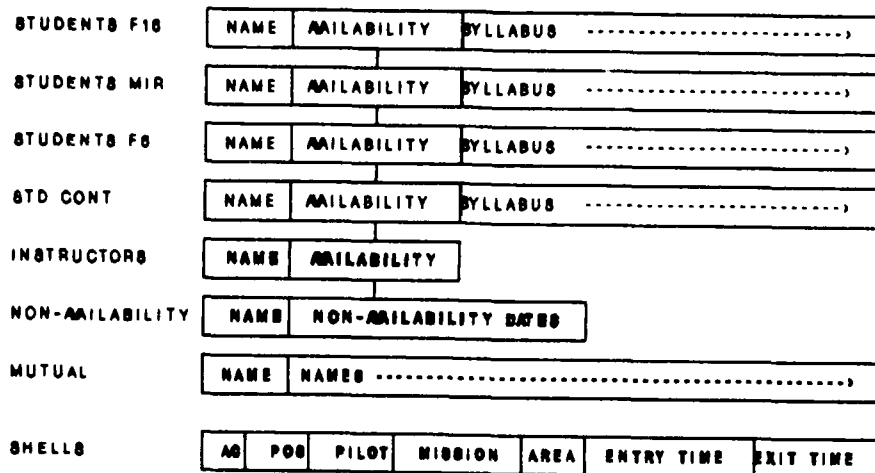


Figure 4.1 data relations

DIALOG

The DSS was built around the kernel identified during the Phase-I of DSS development process. The DSS is designed for use by a computer novice with very little training required. Help is available at the touch of a button. Essential information is provided on the screen to act as memory aids,

and detailed information can be called whenever required. During actual scheduling when the spreadsheet is being used, the user can activate all macros through function keys or menus called up by the function keys. A one-line explanation at the bottom of the screen helps the user in understanding the menu choice. The user need not remember the actions associated with each function key, because holding the CTRL key displays them at the bottom of the screen. However, no detailed help is available in this environment.

All the data information is stored in different data files except for the macros and the spreadsheet menus. This information is retrieved as and when required by the user. Similarly the storage of the daily schedule also takes place in a data file. This technique was adopted to minimize the disk space and memory requirements.

The DSS provides the information to the user regarding the remaining syllabus of each student. The system also provides the mutual flying record at the same time. This information is provided in two windows. The user can select the class for which he needs information. The user can browse through the displayed information by moving the cursor from one window to another.

The system is capable of providing graphical representation of the displayed schedule. This representation helps the user to insure deconfliction of the schedule.

The actual system differs from the planned storyboards in many places. The differences between actual and planned DSS

are discussed below.

Main Menu The main menu (Fig 4.3) designed in the actual DSS differs from the planned version (Fig 4.2) which was based on the feature chart (see page Fig 3.2) developed for this purpose. This difference resulted due to software limitations. As mentioned in Chapter-III, the intended environment could not be created due to VP-Expert's limitations. Therefore, unrestricted transition between the database and spreadsheet was not possible. The actual menu enables the user to perform all his database operations before switching over to the spreadsheet for actual scheduling. The "Make schedule", and "Alter Schedule" options are provided as sub-options to "proceed to make schedule" option.

MAIN MENU							
MAKE SCHEDULE				ALTER SCHEDULE			
A/C Availability				GROUND SCHEDULE			
Flying Phase				AVAILABILITY			
Formation Shell				Students			
Area/Time				Instructors			
Flying Schedule				STUDENT RECORDS			
DAILY UPDATE				PRINT			
Syllabus				QUIT			
Instructor flying							
F10 HELP		Select the option and RETURN					
F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC

Figure 4.2 PLANNED MAIN MENU

MAIN MENU

1. Perform soft update.
2. Perform permanent update.
3. Undo a soft update.
4. Check/Alter availability.
5. Proceed to make schedule.
6. Data entry.

Select option and hit RETURN

<ESC> for help

Figure 4.3 ACTUAL MAIN MENU

The student record option could not be built due to time constraints.

Make Schedule The actual menu (Fig 4.5) is considerably different from the planned menu (Fig 4.4). The planned menu was designed to accept inputs for building the formation shell. The actual menu does not take the user through the whole input process before building the shell. Rather it provides the formation shell for the phase that the user selects. Thereafter, the user can amend by adding shells from other phases to suit his requirements. This difference resulted due to the approach that relied on storage of generic formation shells in different database files. The ability of

the software to link and join the files was used to construct the shell. This approach caters to frequent syllabus changes made by the school. The user can modify any existing file and store the modified formation shell in a new file. This new file can be used for incorporating the syllabus changes in the scheduler.

Altering Availability The screen display for reviewing the non-available list is almost the same as was planned. However the procedure to input non-availability is considerably different from the storyboards designed for this purpose. The actual display (Fig 4.7) accepts inputs for future dates whereas

MAKE SCHEDULE

1. A/C Availability.
2. Flying Phase.
3. Formation Shell
Area/Time
Flying Schedule

Hit <ESC> to return to Main menu.
Select The Option and Press <ENTER>

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure 4.4 PLANNED MAKE SCHEDULE MENU

MAKE SCHEDULE

PLEASE SELECT THE FLYING PHASE. USE <-- --> KEYS TO SEE ALL
THE PHASE LISTING

1V1SIM 2V2CVC 4V2CVC -->

Fig 4.5 ACTUAL MAKE SCHEDULE MENU

the planned storyboard (Fig 4.6) did not provide this option. Besides this the actual system keeps track of availability for each of the three parts of a flying day. This difference was a result of changing system requirements due to learning process. The author learned from the "surrogate user" about this additional requirement, and therefore decided to incorporate it in the system.

Flying Schedule The actual display for filling the shell is the same as was planned. The storyboard relied only on one window for browsing the syllabus and mutual flying record. The actual system has two windows to display the syllabus and mutual flying record at the same time. The screen size available to

ALTERING AVAILABILITY

Rank	Name	Service No	ADD	DELETE
_____	_____	_____	SICK	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	LEAVE	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	TDY	_____
Enter in ADD column to add to non available list			_____	_____
Enter in delete column for removal.			_____	_____
HIT F6 WHEN DONE			_____	_____

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure 4.6 planned non-availability menu

ALTERING AVAILABILITY

NAME:

CLASS:

yy/mm/dd
yy/mm/dd

Sick but fit for gnd duties

Not available at all

From

From

To

To

Hit END to quit this option
<ESC> for help

Figure 4.7 Actual non-availability menu

the user for making or changing the schedule is slightly smaller, but the advantages of simultaneous display of essential data outweigh the screen size consideration

MENU HIERARCHIES

The actual menu hierarchy is slightly different from the planned menu hierarchy. The Fig 4.8 and Fig 4.9 show the planned and actual menu hierarchies respectively. This difference resulted due to the changes made in the planned storyboards. The inability of the software used to provide free movement from one to another was the second reason for this difference.

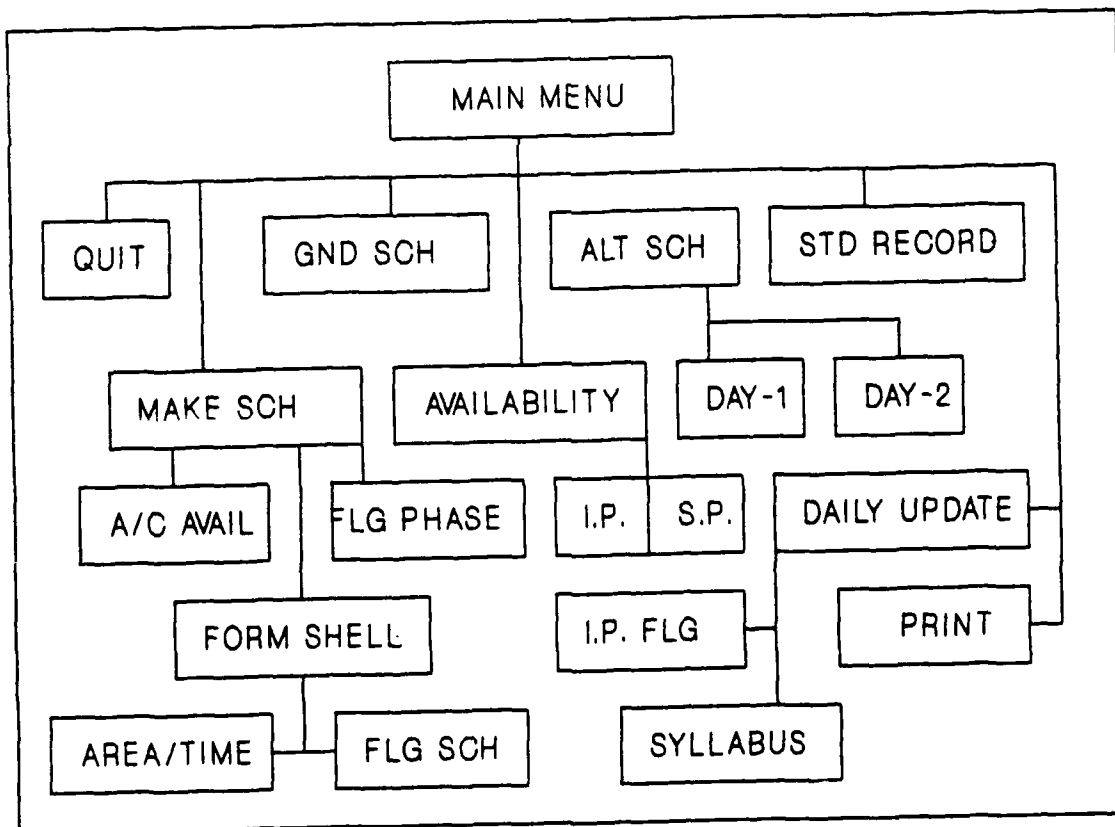


Figure 4.8 Planned menu hierarchy

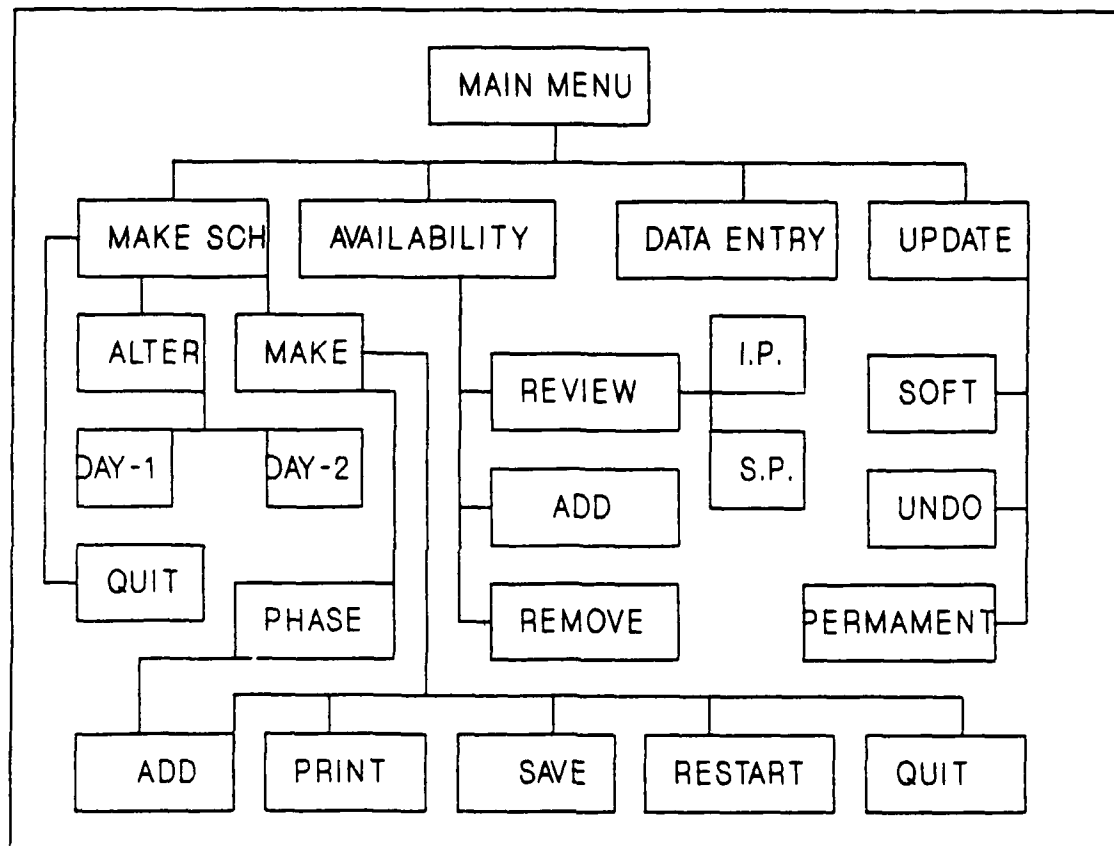


Figure 4.9 Actual menu hierarchy

Chapter V discusses the conclusions and recommendations that resulted from this research and development.

V. CONCLUSIONS AND RECOMMENDATIONS

This thesis developed a scheduling DSS for Combat Commander's School of the Pakistan Air Force. The initial approach envisaged further development of Trapp and Grechanik's (Trapp & Grechanik) model to include the additional requirements. However, this approach was not taken because:

1. The data intensive nature of the problem could not be supported by their model and software.
2. Their system relied on inputs from a wing data file to construct the basic scheduling shell, whereas in this case the shell had to be constructed based on user inputs.

An independent approach was taken to develop the DSS. Concept mapping and storyboards were used to select the kernel for the system. The resulting DSS is an effort for saving time and improving efficiency of the scheduler. The DSS is capable of maintaining and automatic updating of the student syllabus, planned non-availability, and mutual flying records. (The mutual flying records pertain to the number of sorties that a student has flown against another student or with an instructor or with a controller.) The system also allows the user to enter names of the students and instructors at the beginning of a course. The DSS presents all the possible types of formation shells for the scheduler to mix and match for making the day's scheduling shell. The scheduler fills

this shell after consulting the syllabus and mutual flying records, which can be retrieved on the same screen in two different windows. The scheduler can alter a previously made schedule in a similar manner. This DSS is capable of displaying the schedule in a graphical format for an overview of everyone's daily flying engagements. The design and building phase of the DSS were completed without active help of the users. However, the adaptive design approach was followed to facilitate further expansion of the DSS. Implementation of the DSS could not be carried out due to time and distance constraints. The following pages contain the conclusions and recommendations about the specific DSS which are then followed by conclusions and recommendations about DSS and adaptive design in general.

SPECIFIC DSS CONCLUSIONS

This DSS was found to be data intensive in nature. It was seen from this research and review of previous research in this area that a scheduler spends his maximum time on data maintenance. Therefore a major effort went into providing this needed support to the scheduler. The DSS in its present form fulfills the requirements of a scheduler but it has the following limitations:

1. The system stores the accomplished schedules in files for later retrieval but it does not facilitate selective retrieval from these files.

2. The system does not keep record of accomplished ground duties.

3. No provision exists for storing instructor flying hours.

As mentioned in Chapter I, the squadron records are maintained manually. Therefore, this system besides eliminating the need for manual grease boards also opens a path towards total data automation of squadron records. The present system is far from fulfilling the other squadron requirements, but adherence to the adaptive design approach does provide an open end to incorporate further requirements.

MODEL

The scheduling of pilots is done without the aid of a model since no calculations are required to be performed. The optimization of the schedule is also not the aim because the scheduler is only interested in satisfying the requirements. Secondly, frequent alterations to the schedule do not warrant the use of optimization techniques. However, there is room for an Expert System which could take over the scheduling if the scheduler so desires. This system is not required as a replacement to the scheduler. This will be useful when the scheduler has made the crucial parts of the schedule (e.g., assigning leaders and instructors) and then just wants "someone else" to do the rest.

EFFICIENCY VERSES EFFECTIVENESS

This DSS was designed to improve efficiency of the current methods. The system will eliminate the need to maintain and update manual databases. Therefore, it will save a scheduler's precious time. Besides improving efficiency it is likely to improve the effectiveness of scheduling also. As mentioned in Chapter-1, the scheduler often makes mistakes in matching students with instructors, students with students, and students with controllers. This mismatching does influence the student grades and induces an element of bias. Therefore it is believed that improved efficiency will induce improved effectiveness.

RECOMMENDATIONS

The following additions to the DSS are recommended.

1. Data management component of this DSS be developed to facilitate selective retrieval of past flying details. The scheduler often needs to browse through a particular student's past flying details in terms of which instructor he flew with or which controller controlled his formation. The system does store accomplished flying schedules for posterity but does not allow selective retrieval from it.

2. The system be improved to accept instructor flying hours input and allow retrieval of this information for consultation. This is needed because the scheduler requires cumulative instructor hours for fulfilling the secondary requirement of equal distribution of flying (primary being

matching with students).

3. Maintenance of accomplished ground duties record be included in the system. The present system does not keep track of mobile officer and SOF duty record. This record is needed for insuring equal sharing of these duties.

4. An expert system model component be added to assist the scheduler in filling the formation shell. This model must be capable of filling the vacant formation slots when the scheduler has filled the slots where judgements are required.

ADAPTIVE DESIGN METHODOLOGY

Keeping a developing system open-ended for adopting increasing/changing requirements works whether the user and builder are co-located or not. There is no doubt that the co-location will not let the system and the requirements proceed in opposite directions as close interaction will result in immediate adoption of requirements. Whereas on the other hand no user-builder interaction might leave the system and the requirements too far apart, thereby leaving no room for adoption of new/increased requirements. However, experts from the field who may not be the actual users can help in avoiding this wasteful expenditure of time and effort. In the Air Force, the users of a system keep changing places. A system designed and built with the help of a particular user may not be used once he leaves the unit. On the other hand a system built with the help of varying spectra of users has better chance of survival as it will satisfy the requirements

of a broader category of users. During the building phase of the the DSS these surrogate users play a crucial role. A builder's view of the system that he is building, is restricted to what he has conceived. His view is not likely to expand beyond what he has seen in the requirements determination phase, and therefore, will keep the system requirements fixed at the initial level. He also cannot evaluate (from a users perspective) the system that he is building. His evaluation is likely to be a validation (i.e., it works the way it was supposed to) of the system. The reason for flawed evaluation by the builder may be due to the vision/mind tunneling (i.e., he cannot see beyond what he is thinking) effect that a human being experiences when in pursuit of something, in this case the pursuit being building of system. Secondly human propensity for finding no faults in its own creations may influence his evaluation. Therefore, it is essential that the system be evaluated by the users. However, if no user is available, then someone who has some if not "total" knowledge of the user requirements and does not suffer from a builder's biases should be asked to evaluate the system. The evaluation by these "surrogate users" is likely to help in improving the system.

SOFTWARE SELECTION

Selection of software for the system is an essential part of the development process. The VPTM series software selected for this project was expected to provide free movement from

VP-expert to database and spreadsheet. However soon it was realized that the computer memory does not permit this theoretical environment. VP-expert was found to lack the ability to call the database program due to the MS- DOS barrier of 640K. The older version of this software does permit the user to work in a spreadsheet but leaves only 28K of usable memory. The latest version (version-2.1) has even lost this ability as its own memory requirement has increased. Ironically the software manual does not mention this inability. Rather it lists all the procedures for a task which it is not capable of performing. This software limitation was detected too late for the author to consider another software. The database and spreadsheet were used to build the system. These two separate programs are batched together for smooth transfer from database to spreadsheet. The ability of the spreadsheet to retrieve and store database files proved to be an asset.

USERS AND IMPLEMENTATION

Implementation is the last and essential part of the whole development cycle. A system well built but not implemented properly will not satisfy the purpose for which it was built. Implementation is of special importance when a system is not built with the help of the users. To introduce a system, especially the one which is delivered in a finished form, needs the personal efforts of the developer. A champion user is a good way to start this implementation but in the

long run, the whole organization needs to be motivated. This motivation can be achieved through training and cyclic rotation of the scheduling job among all those who are qualified to do this job.

The flying schedule of a fighter squadron is often altered many times before it is accomplished. This asks for a flexible scheduling system that can accept the changes in the schedule with minimum number of adjustments. This rules out the use of analytical modeling techniques (Chapter-1). A model in itself is not good enough for making the decision and it alone does not provide the total framework within which the problem can be adequately tackled. DSS does provide this framework. DSS is a technique for modeling the decision process. It provides a broad problem solving structure which not only has a model but also interfaces (Dialog) the model and its associated data with the user. Those decisions or problems which are either unstructured or semi-structured and require frequent user interference are prime candidates for DSS. It must be admitted that this research started with a preconceived idea of building an expert system model. But the direction of research totally changed when the concepts of DSS were applied to break down the problem into its sub-components. The primary emphasis shifted to the data and dialog component and it was found that the model was only an added benefit that the user would like to have.

RECOMMENDATIONS

The following recommendations are made for DSS development in general:

1. The requirements determination should be carried out with the help of not one but many users. The requirements so generated will represent a broad category of users, and will help in better implementation of the system when there is no one fixed user.

2. In the absence of users, evaluation of DSS be carried out by surrogate users to overcome the builders conceptual and other biases.

3. The selection of a DSS generator is an essential part of the DSS development process. Therefore, the builder must carry out in-depth testing of the software to verify all its capabilities and claims before committing the resources.

4. Implementation of DSS should include plans to train the user organization at large through cyclic rotation of users. This will eliminate the need to recruit a new champion once the old one leaves.

5. An open ended design is essential for successful development of the system. Therefore, adaptive design approach must be followed especially when the users are not frequently accessible.

This scheduling system developed by this thesis is likely to improve efficiency and effectiveness of scheduling in Combat Commanders School. The time saving offered by the

system is likely to change the definition of the scheduler's job. The reduced work-load and time saving offered by the system is likely to make the job of a scheduler more attractive to the pilots who take it up as an additional duty.

The development process followed for the accomplishment of this task proved to be an excellent exercise at learning to attack a problem. The insights achieved during requirements determination and building phase proved the point that breaking order out of chaos is the most important aspect of dealing with a problem. Breaking the problem down to it's sub-components before deciding on the path to follow will help in selection of the most appropriate approach for seeking the goal.

APPENDIX A

STORYBOARDS

MAIN MENU

MAIN MENU

MAKE SCHEDULE	ALTER SCHEDULE
A/C Availability	GROUND SCHEDULE
Flying Phase	AVAILABILITY
Formation Shell	Students
Area/Time	Instructors
Flying Schedule	STUDENT RECORDS
DAILY UPDATE	PRINT
Syllabus	QUIT
Instructor flying	

F10 HELP

Select the option and RETURN

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.1 MAIN MENU

This screen lists the program hierarchy. Make schedule option permits the initiation of the scheduling process from the very beginning. whereas, the alter schedule option is designed to allow the user access to a previously made

schedule for alteration purposes. Daily update is selected for updating of databases after a days flying. Alter availability choice allows the user to alter the names of non-available pilots/controllers. Student record option is for reviewing previous flying details of the student. The choices indented to the right are only selectable once its parent option has been selected

The user employs arrow keys to highlight the option. Option is selected by pressing the return key. Choices available under a certain selection are indented to the right.

F10 toggles help on and off.

ESC returns to previous selection.

F1 through F5 and F7, F8 & F9 select the windows listed under make schedule option. Second press of the same function keys or selection of another window deselects the

MAKE SCHEDULE

MAKE SCHEDULE

1. A/C Availability.
2. Flying Phase.
3. Formation Shell
Area/Time
Flying Schedule

Hit <ESC> to return to Main menu.
Select The Option and Press <ENTER>

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.2 MAKE SCHEDULE

This screen follows the main menu once make schedule option is selected. The display presents a menu for entering number of a/c, flying phase and the area codes and their times. These inputs are required for construction of the flying schedule shell. As the formation shell cannot be constructed before NO:1 and NO:2 inputs are entered, the system will sound a beep when choice NO:3 is selected and no inputs regarding a/c and phase exist in the memory. Arrow keys are used to move the highlighted bar from choice to choice. F10 calls for HELP.

A/C AVAILABILITY

A/C AVAILABILITY

Enter the number of a/c available.

F6 MIR F16

HIT F6 WHEN DONE

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.3 A/C AVAILABILITY

This screen format is used to input the number of a/c available for the course flying. This information along with phase information is used for construction of flying shell. The user enters the number of each type of a/c at the prompt. He can use arrow keys to move back and forth. The user can quit this screen by hitting F6.

FLYING PHASE

FLYING PHASE

- | | | |
|------------|-----------|---------------|
| 1. 1V1 SIM | 5. 2V2VIS | 9. SAT |
| 2. 2V2DISS | 6. 2V4 | 10. INTD |
| 3. 2V2CVC | 7. 4V4 | 11. A.RECCE |
| 4. 2V2ECM | 8. SA | 12. AF STRK |
| | | 13. MASS RAID |

Select option and Hit return

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.4 FLYING PHASE

This screen is designed to allow the user to select the flying phase. This selection is necessary for building the formation shell. Secondly, once the schedule is being made the syllabus files called in different windows present the columns containing the selected phase. The user highlights the desired phase and presses enter for selection. The same action also returns the user to the previous screen. The list of all phases is presented as a memory aid. Arrow keys are used to move the light bar.

FORMATION SHELL

FORMATION SHELL

S.NO:	F6	MIR	F16	AREA	ENTRY TIME	EXIT TIME
	4	2		A		
		4	2	B		

Enter the areas and their timing

HIT F6 WHEN DONE

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.5. FORMATION SHELL

This screen is designed to get inputs from the user for construction of the flying shell. The code and timing of area are entered by the user in the appropriate columns. The number of different type of a/c that will fly at one time in that area are displayed by the system based on flying phase and a/c availability inputs. The system builds shell for the schedule based on this information. For non-course flying the user can input the number of a/c under each type. The area code and time is also required to be entered in the appropriate row. Hitting F6 lets the user quit this option.

FLYING PHASE

FLYING SCHEDULE

DATE:

NAME	POS	A/C	MISSION	AREA	ENTRY TIME	EXIT TIME
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____

F6 SYLLABUS

NAME	TR1	TR2	TR3	1V1	2V2DMS
AFS	1				
BFS					

PRESS F6 TO SAVE THE SCHEDULE

Mobile:

F1	F2	F3	F4	F5	F7	F8	F9
F6 SP	MIR SP	F16 SP	SC	IP	SP/IP	SP/SC	SC/IC

Figure A.6 FLYING SCHEDULE

This screen depicts the formation shell with area and their timings filled in. The screen is offset to the left side to make room for the windows on the right side.

The user can call up any of the five windows by pressing the appropriate function key. The user then enters the name of students in the formation slot depending upon the syllabus. Entering a name in the formation shell will result in the same name being highlighted in the called window to remind the

scheduler about previous commitment. The information called in the window can be either the availability and syllabus data or it can be the mutual flying record of the students and instructors. The different function buttons and their functions are listed at the bottom of the screen for easy recall.

Arrow keys are used to move the cursor. Delete and back space key can be used to erase a wrong entry. The user can move the cursor to the selected window by selecting ALT F6 and browse through the window display for selecting the required segment.

ESC key abandons this screen but a warning is sounded for the user to correct an inadvertent press of ESC key. F6 will save the schedule to a file.

DAILY UPDATE

DAILY UPDATE

DAY-1.SCH

NAME	POS	MISSION	A/C
.....
.....
.....
.....
.....
.....
.....

HIT F6 TO UPDATE THE DATABASES

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.7 DAILY UPDATE

This screen is designed for daily update of syllabus and mutual flying databases. The user is shown the accomplished schedule and is asked to correct it to match the actual flying done (i.e., Remove the names of those pilots who aborted their missions). Once the user is satisfied he can hit F6 to allow the system to do the rest. This action also prompts the user to enter a filename for storing this program. The arrow keys are used to move the cursor around the screen.

INSTRUCTOR FLYING

INSTRUCTOR FLYING

NAME	DATE	FLYING TIME
.....
.....
.....
.....
.....
.....
.....

HIT F6 WHEN DONE

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.8 INSTRUCTOR FLYING

Thus screen is designed for receiving instructor flying inputs. the user is prompted to fill the cumulative flying for the day in front of the listed names. Only the list of those instructors is provided whose names appear on the DAY-1's schedule.

ALTER SCHEDULE

ALTER SCHEDULE

Select The Schedule Which You Want To Alter.

DAY-1

DAY-2

Hit return to make the selection

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.9 Alter Schedule

The display prompts the user to select one of the two existing schedules. After the selection is made the selected schedule is called on to the screen. The operations to be performed on this screen are similar to what the user does on flying schedule screen except that now instead of making a new schedule, he alters an existing schedule. All the information called, in different widows of flying schedule screen is available on this screen also.

GROUND SCHEDULE

GROUND SCHEDULE

MOBILE	FROM	TO
.....
.....
SOF		
.....
.....
.....
RSO		
.....
.....
.....
HIT F6 TO QUIT		

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.10 GROUND SCHEDULE

This screen is designed to accept scheduling of ground duties. Up to three slots can be filled for any day's schedule. The screen is offset to the left to make place for windows on the right side. The user can call any of the syllabus/availability files in the windows by pressing the appropriate function key. These files also contain the information about accumulative ground duties that a person has

carried out so far. This schedule is always made after the flying schedule. Any conflict with the flying schedule is indicated by a beep and rejection of selected name by the system. The user can view the flying and ground schedule on a deconfliction graph by pressing ALT F1.

The user is prompted to enter the timing first so that deconfliction with the flying schedule can be done by the system.

Arrow keys are used to move the cursor around. F6 saves the schedule and F10 calls for help. ESC key abandons the process. However, a warning is displayed to save the user from accidental press of ESC button.

NON-AVAILABILITY STATUS

NON-AVAILABILITY STATUS
STUDENTS

F6	MIR	F16	CONTROLLERS
.....
.....
.....
.....
.....
.....

Do you wish to alter the list (Y/N).

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.11 Non-Availability

This screen lists the names of students who are not available on any given day. The names are listed under the class headings. This screen is for viewing purposes only. The user can either select to alter this list by typing "Y" at the prompt or return to make schedule menu by typing "N".

ALTERING AVAILABILITY

ALTERING AVAILABILITY

Rank	Name	Service No	ADD	DELETE
.....	SICK
.....
.....
.....	LEAVE
.....
.....
.....	TDY

Enter in ADD column to add to non available list
Enter in delete column for removal.
HIT F6 WHEN DONE

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.12 ALTERING AVAILABILITY

This screen is used to update the availability list of students. The names of students can be added to or deleted from the list. For addition to the list, the user types the name of the student in the ADD column in front of the appropriate heading. Similarly the names to be deleted are typed in the delete column. Left side of the screen lists the current non-available students list which is continuously updated as return is hit after each entry. Once the user is finished he can quit this screen by hitting F6.

STUDENT RECORD

STUDENT FLYING RECORD

NAME:

DATE	MSN TYPE	POS	INSTRUCTOR
.....
.....
.....
.....
.....
.....

HIT F6 TO QUIT

F1 F6 SP	F2 MIR SP	F3 F16 SP	F4 SC	F5 IP	F7 SP/IP	F8 SP/SC	F9 SC/IC
-------------	--------------	--------------	----------	----------	-------------	-------------	-------------

Figure A.13 STUDENT RECORD

This screen represents student record of past course flying. This screen is initially blank. However, once the user enters a name, past fifteen days flying information about that student is displayed. Hitting ENTER clears the screen and the user is prompted to enter the next name. Any time the user wants to quit, he can do so by hitting F6.

APPENDIX B

USERS MANUAL

The purpose of this manual is to familiarize the first time user with the operation and control of this program.

PROGRAM INSTALLATION

This program can be used on computers equipped with two disk drives or a single high capacity drive or a fixed drive. The operating speed of the program is slower when no fixed drive is used.

Fixed Drive Installation Make a directory named VPI and copy the VP-Info operating system and all the files from program diskette except for Autovp.wks to it. The file, named VPI.CNF provided on the program diskette is modified to facilitate starting the program from the MS-DOS prompt. The batch file named CCS.BAT is included for transferring program control to VP-Planner as and when required.

Make a second directory named VPP and copy the VP-Planner program to it. Make a sub-directory called WKS in VPP and copy the file named AUTOVP.wks to it. Your program is now ready for use. Change to directory c:\VPI and type "CCS" to start the system.

MAIN MENU

The menu displayed to the user will be as shown in Figure B.1

MAIN MENU

1. Perform soft update.
2. Perform permanent update.
3. Undo a soft update.
4. Check/Alter availability.
5. Proceed to make schedule.
6. Data entry.

Select option and hit RETURN

<ESC> for help

Figure B.1 Main Menu

Soft update

This option permits the user to temporarily update the syllabus and mutual flying databases based on DAY-1's flying program. This action is reversible. This option should be used when DAY-2's program is to be made and DAY-1's scheduled flying has not been accomplished. The temporary update allows the user to see the syllabus and mutual flying records as if all the scheduled flying has been done. In case the user tries to perform two consecutive updates, the system displays an error message and returns the user to main menu.

Undo Soft Update

This option is meant to reverse the effects of soft update. If undo option is selected with no soft update performed earlier on, the system displays a message to this effect and returns the user to main menu.

CAUTION: Do not make amendments to the day-1's schedule after performing a soft update. If you have performed a soft update you must undo it.

Permanent update

This option is designed to be used only when DAY-1's flying has been accomplished. The selection of this option will update the Syllabus and Availability databases like the soft update option. However, this action is irreversible.

Data Entry

This option is designed for entering or changing the names of pilots and controllers as and when required. If this option is used to enter the names of new arrivals at the beginning of a course, it is essential to copy the following files from the original program diskette.

F6-std.dbf, MIR-std.dbf, F16-std.dbf. SC.dbf, IP.dbf,
and VERSIS.dbf

This will reset the remaining syllabus to the stipulated values. The selection of this option takes the user to a sub-menu which is shown in Figure B.2. The user can select either of the options to see and alter the names. The display will show a list of names which can be replaced altered. The user can move the cursor in the vertical axis

using Pg Up or Pg Dn keys. Hitting return will update the databases and user will be returned to main menu. These actions will alter the names listed in the vertical columns of data tables. However, to change the names in the horizontal axis of mutual flying database, a macro command needs to be executed while in the spreadsheet environment. Changing of records will be completed once "<ALT> A" is pressed during spreadsheet operations. Selection of QUIT returns to main menu.

DATA ENTRY

1. Enter the names of F6 class.
2. Enter the names of MIR class
3. Enter the names of F16 class.
4. Enter the names of controllers.
5. Enter the names of instructors.

Select option and hit RETURN

<ESC> for help

Figure B.2 Data entry

CAUTION: Do not select any update options till changing of records has not been totally completed.

Check/Alter Availability

Selection of this option displays the Check/Alter Availability sub-menu. This sub-menu is shown in Figure B.3

AVAILABILITY STATUS

1. Review students availability.
2. Review instructors availability.
3. Remove from non-availability list.
4. Add to non-availability list.
5. QUIT

Select option and hit RETURN

<ESC> for help

Figure B.3 Check/Alter Availability

Review Student Availability This option is designed for the user to review the list of non-available students for a particular date. The selection of this option will result in a prompt questioning the user about the date. The date input in yy/mm/dd format will result in display of non-available students list. This display will show the non-availability of each listed student in each of the three parts of a flying

day. An 'F' in the column "First", "Second", or "Night" tells that the respective student is not available in that particular part of the day. Whereas a 'T' in the said columns means otherwise. Hitting any key will return the user to parent menu.

Review Instructor Availability This option operates in the same manner as the review student availability option.

Add to Non-Availability List This option is designed to accept user inputs about expected non-availability. The screen display presented to the user is shown in Figure B.4

ALTERING AVAILABILITY

NAME:

CLASS:

yy/mm/dd

yy/mm/dd

Sick but fit for gnd duties

From

To

Not available at all

From

To

Hit END to quit this option
<ESC> for help

Figure B.4 Altering Availability

The user can enter the name and the system will check to return the class information. If user now changes the class

information, the system again checks to insure that the name exists in the databases. This option to rewrite the class name is provided to cater to pilots/controllers with same names. However, the system is not capable of sorting duplicate names belonging to the same class. The user then enters the dates of non-availability and answers the prompted questions. This information is stored in a file for daily updates of availability status. Hitting END will return the user to parent menu.

Removal From Non-Availability List If the user wants to amend or delete the information added during ADD to non-availability operation, he can select this option to review the file listing. The names can be deleted by using back space or Delete keys. Deletion of name only will result in complete deletion of the entry by the system. To alter information date entry format should be adhered to. It is advised that instead of making an alteration, complete entry should be deleted and then added with the add option (option 4). Selection of QUIT returns to main menu.

PROCEED TO MAKE SCHEDULE

This option should be selected only when the data operations are finished. The system will have to be restarted to return to main menu once this option has been exercised. The execution of this option takes the user into spreadsheet environment for actual scheduling operations. However, before moving to spreadsheet, the user is prompted to give the date

of schedule. This information is required for insuring that only information regarding non-availability on the day of schedule is presented. CAUTION: If you are using dual disk drives, change the diskette in drive a: to VPP-Planner program diskette before selecting this option.

SCHEDULE

As soon as the loading of program is completed, a menu appears on the the left bottom of the screen. The instructions to make the selection are displayed on the screen. The schedule menu options presented are:

MAKE ALTER QUIT

A one line explanation below the highlighted choice is also displayed.

MAKE

Selection of this option will result in display of another menu asking the user to select the flying phase. Arrow keys are used to browse through the available choices. Selection of any one of these menu choices will retrieve the formation shell for that phase. Before making any entries in this shell it is advisable to complete the flying shell for the day. To do this press CTRL F6. This action will display another menu at the bottom of the screen. The menu options are explained one by one.

AREA/TIME This option allows the user to enter area codes and their entry/exit times. This option will prompt the user regarding each entry. However if the user desires, he may enter the information in the respective columns without

exercising this option. The only caution to be observed is that no entry should be made in the first row of "AREA", "FROM" and "TO" columns.

ADD Selection of this option will display instructions to add more formation shells to the current shell. The user is required to move the cursor below the last formation in the column titled "PILOT". Pressing CTRL F7 will add the new shells to the current one. This action can be repeated as many times as the user desires.

SAVE This option when selected will save the program to the diskette. However, before that it will ask the user to select the name of the file. The only options available are DAY-1 or DAY-2. Do not save the program as DAY-1 if DAY-1's program already exists as this action will wipe out the previous program.

RESTART Selection of this option returns the user to schedule menu. The current screen is erased and user can start from the beginning again.

QUIT This selection will result in system quitting the program. The user will be returned to MS-DOS prompt.

ALTER

This option when selected from the schedule menu will display the two schedules. (DAY-1 and DAY-2). The user can select any one of these and make amendments to it. The sub-menu options available under CTRL F6 can also be used during alterations.

DATA RETRIEVAL

The system offers the needed data retrieval capability at the touch of a button. Either while making or amending a schedule, the user can see student availability, syllabus and mutual flying records. This information is simultaneously displayed in two windows. This facilitates browsing through this information with the concerning part of schedule visible in the top left portion of the screen. The figure B.5 shows the window distribution and names the information displayed in each window.

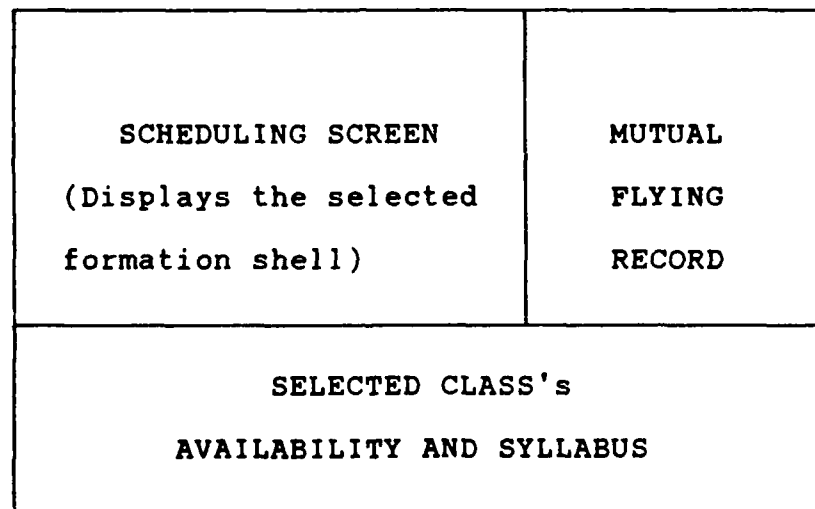


Figure B.5 Window Displays

The information in each window is retrieved through the function buttons listed below

- F1 F6 student's availability/syllabus.
- F2 MIRstudent's availability/syllabus.
- F3 F16 student's availability/syllabus.

- F4 Student Controller's availability/syllabus.
- F5 Instructor Pilot's/Controller's availability.

Pressing CTRL and any of these function buttons will retrieve the listed information. This action will also retrieve the mutual flying record. user can move the cursor from one window to another by hitting F6. Pressing CTRL F9 clears the windows.

GRAPHICAL DISPLAY OF SCHEDULE

The system has the provision to display the schedule graphically. This option can be activated by pressing CTRL F8. To return to scheduling screen Press CTRL F9.

BIBLIOGRAPHY

1. Baker, Kenneth R., Introduction To Scheduling And Sequencing New York: John Wiley And Sons INC, 1974.
2. Beightler, Charles S. ; Phillips, Don T. ; Wilde, Douglass J., Foundations Of Optimization(Second Edition) New Jersey: Prentice Hall INC, 1979.
3. Bodin, L; B. Golden; A. Assad and M. Ball. "Routing And Scheduling Of Vehicle And Crews: The State Of The Art" Computational Operations Research, 10: 62-212 (1983).
4. Davis, Michael W. Applied Decision Support. Englewood Cliffs, NJ: Prentice Hall Inc, 1988.
5. Dervitsiotis, Kostas N., Operations Management New York: McGraw-Hill Book Company, 1987.
6. Dyer, Capt Doug; Walters, Ms Sharon, "Aircrew Scheduling: An Application Of Expert System Technology" Presented at: Symposium On Command And Control Research, (1989)
7. Drowley, Capt R.D. Computer Assisted Flight Schedule Optimization. MS Thesis, Army Command And General Staff College, Fort Leavenworth KS. (June 1984).
8. Hamon, Jean-Paul; Weissert, Francois. " ROSSINI: A General Model For Optimizing Aircraft Assignment To Parking Positions At An Airport", Presented at: Tenth International Conference On Operational Research.
9. Hogarth, Robin H. Judgement And Choice: A Psychology of Decision. John Wiley and Sons Inc.
10. Holst, Olav. "Combined Scheduling And Maintenance Planning For An Aircraft Fleet", Presented at: Tenth International Conference On Operational Research.
11. Honour, Craig Gibson A Computer Solution To The Daily Flight Schedule Problem MS thesis. Naval Postgraduate School Monterey CA (June, 1975).

12. Huelskamp, Robert M., Aiding USAF/UPT(undegraduate pilot training) Aircrew Scheduling Using Network Flow Models MS Thesis. School of Engineering, Air force Institute Of Technology, Wright-Patterson AFB OH (June 1986).
13. Keen, Peter G. W. and Michael S. Scott Morten. DECISION SUPPORT SYSTEMS An Organizational Perspsctive. Addison Wesley Publishing Company, Inc. 1978.
14. Kopf, Thomas J., Design Of An Aircrew Scheduling Aid For The 6916th Electronic Security Squadron MS Thesis, School Of Engineering, Air force Institute Of Technology, Wright-Patterson AFB OH (June 1987)
15. Mathews, Mark T., A Training Management And Scheduling System For USAF Tactical Fighter Squadrons MS Thesis, School Of Engineering, Air Force Institute Of Technology, Wright-Patterson AFB OH (June 1987).
16. Mortenson, R.E., "Maintenance Planning And Scheduling Using Network Simulations", Winter Simulation Conference Report, Vol 1 333-340 Atlanta Georgia (Dec 1981).
17. Pannel, Carlton L., A Linear Programming Application To Aircrew Scheduling, MS Thesis, Army Command And General Staff college , Fort Leavenworth, KS (Jun 1980).
18. Robey, Daniel and Taggart, William "Human Information Processing In Information And Decision Support Systems" DSS: A Data Based, Model Oriented, User Developed Discipline edited by: William C. House. Petrocelli Books Inc, 1983.
19. Roege, WilliamH., Pilot Scheduling In A Fighter Squadron MS Thesis, School Of Engineering, Air Force Institute Of Technology, Wright-Patterson AFB OH. (Feb 1983).
20. Sahli, Gregory J. & Shacklett, Jack L., A Prototype Microcomputer Decision Support System For Aircrew Scheduling MS Thesis, School Of Engineering, Air Force Institute Of Technology, Wright-Patterson AFB OH (Sep (1983)

- 21 Shreuder, J.A.M., "Timetables In Dutch High Schools",
Presented At: Tenth International Conference On
Operational Research
22. Simon, Herbert A. The New Science Of Management Decision.
New york: Harper and Row, 1960.
23. Sprague, Ralph H. and Eric D. Carlson. Building Effective
Decision Support Systems. Englewood Cliffs, NJ: Prentice
Hall Inc, 1982.
24. Trapp, Paul E.; Grechanik, Jeffery W., "Design Evolution
Of A Fighter Training Scheduling DSS" MS Thesis, School
Of Engineering, Air Force Institute Of Technology,
Wright-Patterson AFB OH (1987).
- 25 Valusek, LT.COL. John R. "Adaptive Design of DSS's: A
User Perspective". DSS 88 TRANSACTIONS, 1988.
26. Young, Lawrence F. Decision Support and Idea Processing
Systems. Dubuque, Iowa: Wm. C Brown Publishers, 1989.

Vita

Javaid Anwar [REDACTED],

[REDACTED]. He graduated from Pakisatn Air Force College with Bachelor of Science degree in Mathematics/Physics in 1975. He joined Pakistan Air Force Academy in the same year, and was commissioned in March, 1977. He completed his operational conversion to fighters in March, 1978. He has served in various fighter squadrons of Pakistan Air Force. His assignments have ranged form a Squadron pilot to a Flight Commander. In August of 1988 he entered the School of Engineering, Air Force Institute of Technology (AFIT), Wright Patterson AFB, Ohio.

[REDACTED]

[REDACTED]

[REDACTED]

[REDACTED]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION Unclassified			1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution unlimited		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/COR/ENS/90M-2			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION School of Engineering		6b. OFFICE SYMBOL (If applicable) AFIT/ENS	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State, and ZIP Code) AIR FORCE INSTITUTE OF TECHNOLOGY Wright Patterson AFB, Ohio 45433			7b. ADDRESS (City, State, and ZIP Code)		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO	PROJECT NO	TASK NO
			WORK UNIT ACCESSION NO		
11. TITLE (Include Security Classification) A DECISION SUPPORT SYSTEM FOR SCHEDULING IN COMBAT COMMANDERS SCHOOL					
12 PERSONAL AUTHOR(S) Javaid Anwar, BSc., Squadron Leader Pakistan Air Force.					
13a. TYPE OF REPORT MS Thesis		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 1990 March	
15. PAGE COUNT 100					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Decision Support System, Adaptive Design, Scheduling, Multi-Squadron Scheduling.		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>Thesis Advisor: John R. Valusek, PhD, Lt Col, USAF Assistant Professor of Operations Research.</p> <p>(See reverse for Abstract)</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL John R. Valusek, Lt Col, USAF			22b. TELEPHONE (Include Area Code) (513) 255-2549		22c. OFFICE SYMBOL AFIT/ENS